Controlling Japanese Machine Translation Output by Using JLPT Vocabulary Levels

Alberto Poncelas and **Ohnmar Htun**
Rakuten Institute of Technology
Rakuten Group, Inc.
{alberto.poncelas, ohnmar.htun}@rakuten.com

Abstract

In Neural Machine Translation (NMT) systems, there is generally little control over the lexicon of the output. Consequently, the translated output may be too difficult for certain audiences. For example, for people with limited knowledge of the language, vocabulary is a major impediment to understanding a text.

In this work, we build a complexity-controllable NMT for English-to-Japanese translations. More particularly, we aim to modulate the difficulty of the translation in terms of not only the vocabulary but also the use of kanji. For achieving this, we follow a sentence-tagging approach to influence the output.

1 Introduction

In the Natural Language Processing research, text simplification aims to find variants of a text which convey the same meaning but are expressed in a simpler form. This process includes modifications such as reducing the length, decreasing the use of infrequent words, etc. Simplification systems are useful for helping certain populations such as children, non-native speakers, and people with a low level of literacy or language disorders (Štajner and Popović, 2016).

In this work, we apply simplification to the translation task. In particular, we aim to control the lexicon complexity of English-to-Japanese Neural Machine Translation (NMT) models. The output generated by an NMT system in Japanese may be too difficult to understand for a person with more limited knowledge of the language. An example of this is the use of kanji ideograms. Certain kanji are learned in the later stages of education[1], which causes some people not to be entirely familiarized with all of them. This implies that both vocabulary and kanji may represent an accessibility problem. Accordingly, we focus on influencing the output of an NMT system to control whether it should produce more or less difficult words. This can be measured based on the vocabulary lists provided for different levels of the Japanese Language Proficiency Test (JLPT). A more detailed explanation of this can be found in Section 2.

For modulating the translation, our approach is inspired by other works consisting of influencing the generation of sentences for certain domains or languages. This can be achieved by including a tag at the beginning of each sentence stating how the output should be. For our approach, we use tags to indicate the level of lexicon complexity expected in the output. We first insert a token at the beginning of each training sentence according to the complexity of the Japanese target side. Then, at decoding time, we can influence the output by using such tags.

This paper describes such an approach, and explores the following Research Questions (RQ):

RQ1: Can the vocabulary complexity of the output be controlled adding tags in the source sentences?

The addition of tags in the source sentences to control the output of the NMT models has been explored not only for different domains (Chu et al., 2017) but also for different languages (Johnson et al., 2017). We want to explore these techniques for Japanese translation and investigate how could it be used to control the complexity level of the output.

RQ2: How much does the output level agree with the complexity level indicated in the input?

Although adding tags could bias the complexity of the translation, it has limitations. For example, some translations may require the use of complex vocabulary despite the restrictions. We analyze to what extent the complexity of the sentences generated by the NMT corresponds to those indicated in the input.

[1] https://en.wikipedia.org/wiki/Ky%C5%8Diku_kanji

Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022), pages 77 - 85

RQ3: How much does the restrictions in complexity impact the translation quality?

Introducing tags to restrict the complexity could lead also to degradation of the performance of the NMT in terms of adequacy. Our third research question aims to investigate how much these restrictions impact the translation.

2 Japanese Language and JLPT

The Japanese language has three writing systems[2]: hiragana (46 characters); katakana (46 characters); and kanji (more than 2000 characters).

Hiragana is mainly used for native Japanese words whereas katakana is used for foreign words or onomatopoeia. For example, the translation for the word "hat" is ぼうし (read as "boushi") which is written in hiragana. Alternatively, some people may use the term borrowed from English "ハット" ("hatto") which is a transliteration of "hat". As it is a loanword, it is written in the katakana syllabary.

Despite that, native Japanese speakers would use more frequently the kanji ideogram 帽子 (which is also read as "boushi") for "hat".

Although it is possible to fully express in Japanese using hiragana or katakana exclusively, kanji is usually used. Despite that, as there exists more than 2000 kanji, a Japanese learner would assimilate them gradually, and therefore be more comfortable using hiragana for writing or reading certain words.

A popular criterion to measure the level of proficiency in Japanese for non-native speakers is the Japanese Language Proficiency Test (JLPT). It is a five-level grading system that ranges from JLPT 5 (the most basic) to JLPT 1 (the most advanced). These five levels are also referred to as N5, N4, N3, N2, and N1.[3]

In this work we use both notations, "JLPT" or "N" , indistinctly. Additionally, we may refer to *higher levels* to those JLPT levels closer to N1, and *lower levels* to those closer to N5.

3 Related Work

In the text simplification field, several approaches alter the complexity of the lexicon. For example, Glavaš and Štajner (2015) propose replacing difficult words with a simpler synonym. Furthermore, Hading et al. (2016) perform the complex-word replacement applied for Japanese language.

Alternatively, Wang et al. (2016) build a monolingual NMT system to transform sentences into a simplified version in the same language.

Nishihara et al. (2019) propose a similar monolingual sequence-to-sequence system with several levels of complexity in English. These are based on the grade level of US education system. Similarly to our work, they control the complexity by using special tokens for each grade.

Performing text simplification in combination with translation has also been explored by Štajner and Popović (2019). They focus on using automatically simplified sentences as the input of an NMT model.

Regarding the complexity-controllable translation, Spring et al. (2021) aim to produce translations based on different levels established by the Common European Framework of Reference for Languages (CEFR).

Shardlow and Alva-Manchego (2022) also performs combinations of simplification and translation (*Translate then Simplify*, *Simplify then Translate* and *Direct*) to generate simplified translations.

There are previous works that use tags to control the output. Martin et al. (2020) extract different characteristics that measure the complexity and include them as tags in the source to condition the output. Similarly, Agrawal and Carpuat (2019) also use a tagging system, training the model with a dataset where the same sentences have been rewritten at different complexity level. Finally, Marchisio et al. (2019) use two tags (i.e. "simple" and "complex") to classify the sentences by difficulty.

Some difference with our research is that we use a five-tag system based on the JLPT framework. In addition, as we explore the Japanese language, the definition of complexity also considers spelling. Therefore, depending on the writing system, some words may have different complexity levels.

4 Complexity-Controllable Translation

Our proposal consists of building an English-to-Japanese NMT model with a controllable lexicon complexity. In this work, the *complexity* is measured based solely on the vocabulary of the different JLPT levels.

There are two main processes involved: (i) determine the JLPT level of a sentence (Section 4.1); and (ii), include the complexity level in the training

[2] https://en.wikipedia.org/wiki/Japanese_writing_system

[3] This notation comes from the first letter of Japanese name of the JLPT, "Nihongo Nōryoku Shiken"

Word	JLPT level
友達	3
ともだち	5

Table 1: Example of the word mapping. Each word is assigned a JLPT level. Although both words convey the same meaning, they belong to different JLPT levels due to the writing system.

process of the NMT model (Section 4.2).

4.1 Sentence Classification

Initially, we build a classifier to estimate what is the JLPT level of a sentence. Following the proposal of Ramkissoon, a sentence can be classified with the level of that of the most difficult (highest level) word of the sentence. This approach assumes that one can understand a sentence if one is capable of understanding each word. This is not necessarily true, as usually there are other components involved such as the length of the sentence, the grammar, or the number of clauses. For future work, we propose to expand this assumption of complexity and include a more detailed classification.

Deciding the level of a word can be done based on the vocabulary lists of JLPT levels. We use the resources from Waller (2010)[4].

For each JLPT level, we obtain a list of words and a list of kanji that a Japanese student should be familiar with. We combine this information to build a mapping between each word and its JLPT level as in Table 1. This map also takes into consideration the spelling of the words as follows:

- The word is spelled using hiragana: Its corresponding JLPT level will be that of the vocabulary list.
- The word is spelled using kanji: The JLPT level of this word is that of the level of the most difficult kanji.

Note that the same word can be classified as two different levels depending on the spelling. For example, the words we see in Table 1, 友達 and ともだち, are both the same word ("tomodachi", which means "friend" in English). They have different JLPT levels because ともだち only contains hiragana which is readable by the lowest levels of fluency (JLPT 5) whereas the word 友達 is formed by the kanji 友 (JLPT 5) and 達 (JLPT 3), and therefore that word is categorized as JLPT 3.

Considering a sentence t a sequence of words $(t_1, ..., t_{|t|})$, the JLPT level of the sentence will be that of the word t_i with the highest difficulty according to the mapping. If a word is not in the mapping, such as an English or an out-of-vocabulary term, we assume it is a proper name and it will be ignored (equivalent to assuming that it is in the level JLPT 5).

4.2 Machine Translation Training

The models we build should generate translations biased towards the complexity levels established in the input. The method we follow is by adding a complexity tag to the sentences.

Including a special token in the source to control the output of an NMT technique has demonstrated good results for translating into different domains (Chu et al., 2017) or even into different languages (Johnson et al., 2017).

This technique consists of preprocessing each sentence pair (s, t) in the training and dev set as follows:

1. Classify the Japanese sentence t as described in Section 4.1 and retrieve the JLPT level l.
2. Build a token N_l according to the level l. To avoid using just numeric values our tag consists of concatenating the letter N with the level together. For example, the token N_l for JLPT 1 we build would be "$N1$".
3. Expand the English source-side sentence by adding the token in the beginning $s' = (N_l, s_1, ..., s_{|s|})$.
4. Retrieve the pair with the expanded source (s', t).

The processed data is used to train an NMT model. By doing this, the system should learn the relation between the first token in the source and the vocabulary on the target side. Later, at decoding time, we include a tag with the desired JLPT level so the model should generate sentences including the vocabulary of such level.

5 Experiments

We build NMT models in the English-to-Japanese direction using Marian NMT (Junczys-Dowmunt

[4] https://www.tanos.co.uk/jlpt/

et al., 2018). These models consist of a transformer (Vaswani et al., 2017) model with 6 layers both in the encoder and 6 in the decoder. We train it for a maximum of 500K steps (18 epochs).

We use one of the biggest English-Japanese corpus, JParaCrawl v3.0 (Morishita et al., 2020), as train set (25.7M sentences) and 10K randomly-selected sentences from Tatoeba (Tiedemann, 2012) as dev set.

For the experiments, we build two models. One model is built with plain data without any modification that serves as a reference for comparison purposes. The second model is built by including tags as described in Section 4.2. We use kytea (Neubig et al., 2011) to split sentences and extract the vocabulary of the Japanese side.

For testing the models, we randomly selected 5000 sentences from Tatoeba (from those not included in the dev set). This dataset is built for educational purposes and therefore there are sentences of different complexities.

First, we translate these sentences with the plain model. Then, for the model that uses tags, we replicated each sentence five times and added a different tag (from N1 to N5) to each of them. By doing this we encourage the model to produce translations of different levels for each input.

This means that we generate six alternative translations from a single test set. One output is the translation of the plain NMT model trained without tags ("no-tag" output). The other five outputs correspond to the translation when one of the tags is added at the beginning of the sentence. In the following, we name each output with the tag added in the source. For example, we refer as *N2 output* to the translations when the tag "N2" was added in the input sentences.

6 Experimental Results

We divide the analysis of the results of the experiments into four different sections: (i) Section 6.1, where we explore the simplification capabilities of the NMT model (RQ1); (ii) Section 6.2, where we analyze the agreement between the output level and that stated in the input (RQ2); (iii) Section 6.3, where we investigate the translation quality (RQ3); and (iv), Section 6.4, where we provide translation examples that illustrate the effect of constraining complexity in the output.

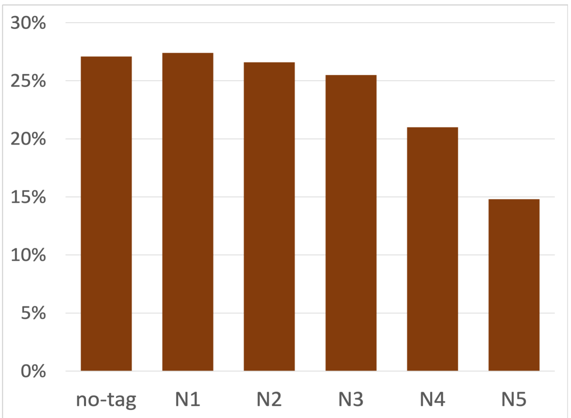

Figure 1: Average percentage of kanji

6.1 RQ1: Can the vocabulary complexity of the output be controlled adding tags in the source sentences?

The tags used to modulate the complexity are based on the vocabulary and kanji in Japanese. Accordingly, we explore whether the outputs of the model are simplified in terms of these.

First, we explore the usage of kanji. For someone with limited knowledge of Japanese, it is expected the kanji comprehension to be lower also. Therefore, the outputs in the lower levels should contain a smaller proportion of kanji. In Figure 1 we present what is the proportion of kanji in the outputs. In the reference 27.5% of the characters are kanji, which is similar to the output of the NMT model with no tags. This is also the proportion in the outputs of higher levels of JLPT (in fact, the N1 output has a slightly higher usage of kanji than the plain model).

The proportion of kanji decreases gradually as lower JLPT levels are stated in the input. For the N5 output, the percentage of characters that are kanji is just 14.8%. Therefore we can say that in terms of kanji usage, the inclusion of tags is beneficial to decrease the complexity.

In addition to that, we compare the vocabulary sizes of the translations. In general, the more restricted the generation is, the lower the size of the vocabulary is expected to be. In Figure 2 we present the number of distinct words in each output.

In the plot, we see that the size of the vocabulary for the model with no tags is similar to those of higher levels such as N1 or N2. The number of words tends to decrease the more restricted the complexity is.

The N5 output seems to be an exception to that,

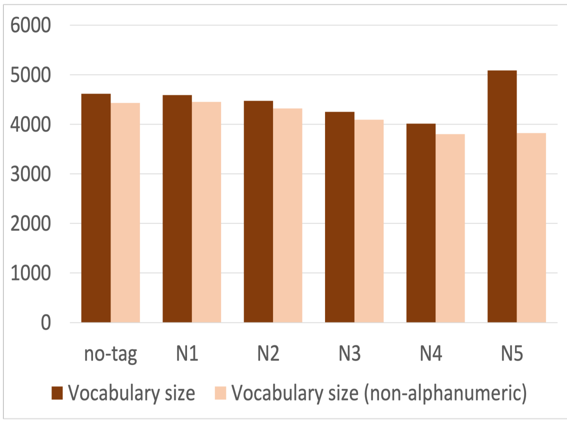

Figure 2: Vocabulary size of the output.

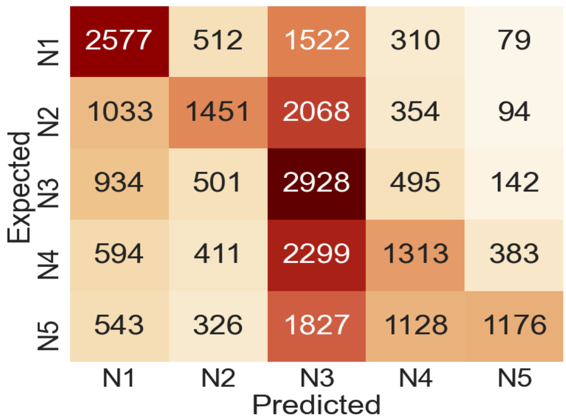

Figure 3: Confusion matrix of the classification of the output

as the vocabulary size exceeds that of the N1 output. However, upon inspection of the translations, we discovered that many words were just copied directly from the source instead of being a translation. We decided to include in that plot the size of vocabulary after removing the alphanumeric terms (e.g. English words, numbers) from the output as it may distort the analysis. Under these circumstances, we observe that the number of words also decreases for the N5 output. In this case, the vocabularies of the translations range from 4400 words (for less restricted outputs such as no-tag or N1) to 3800 words (for N5 output).

Consequently, the sizes of the vocabularies also indicate that the complexity tag is useful to limit the diversity of words.

6.2 RQ2: How much does the output level agree with the complexity level indicated in the input?

For answering the second RQ, we want to estimate whether the outputs match the levels stated on the

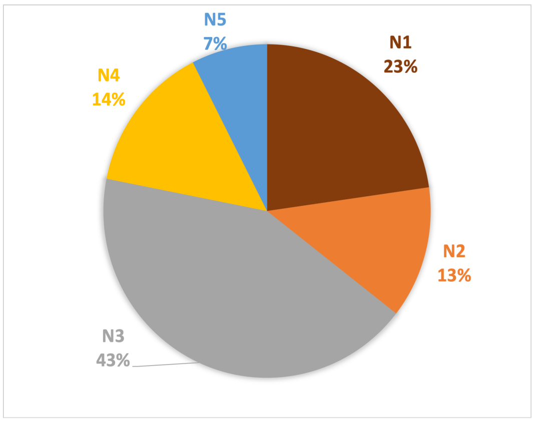

Figure 4: Proportion of sentences of each JLPT level (output file).

test set. Therefore, we classified the outputs of the model (25000 sentences) and compare them to the level that was prepended as a tag on the source side. In Figure 3 we present the heatmap of the confusion matrix.

We found that only 38% of the sentences in the output had an exact match with the proposed tag on their source side. Most of the sentences in the output of the model are predicted to be JLPT 3 level as can be seen in Figure 4.

In addition, we also see that many disagreements occur in sentences where lower complexity is expected. Many sentences in the N5 output include kanji of more advanced levels. We find two main reasons for that.

One reason is that certain terms may not exist in the vocabulary of lower JLPT levels. For example, even when attaching an N5 tag to a sentence, the model may not be able to translate difficult concepts such as “monopolize” or “corruption” that do not exist in the vocabulary of such a low level.

Another reason is that the model does not have enough information to generate an adequate translation. For example, the word “window” is a basic word that would be categorized as JLPT 5 (lowest complexity level) if it is spelled in hiragana as “まど”. However, in the N5 output, this was spelled using the kanji “窓” instead, which is considered to be in level JLPT 3. This occurs because it is unlikely to find the hiragana spelling in most texts. Upon the inspection of the training data, we did not find any occurrence of translation of “window” spelled in hiragana.

Answering this RQ, we find that the model is not very accurate in terms of generating translation in the JLPT level as expected in the input.

Added tag	BLEU (full)	BLEU (hiragana)
no-tag	23.3	24.8
N1	22.4	23.8
N2	21.8	23.1
N3	21.8	22.8
N4	18.9	20.8
N5	13.8	15.2

Table 2: Translation quality of the outputs measured with BLEU metric. The column *BLEU (full)* presents the scores of the output when compared to the original reference. The column *BLEU (hiragana)* presents the scores after both output and reference have been converted into a single writing system (i.e. hiragana).

Alternatively, one may consider that the knowledge of vocabulary should be cumulative. In the experiments, we used the hardest word to tag the sentence, which implies that sentences classified as JLPT 1 or 2, also contain the vocabulary of lower levels. This is coherent with the problem of different literacy levels, as an advanced reader is also capable of reading sentences with simpler vocabulary. In such a case, we could consider that the output should be at either the same or lower level than that stated in the input. Then, the number of correctly classified sentences ascends to 62%. Despite that, the problem of lower-level sentences containing difficult kanji remains.

6.3 RQ3: How much does the restrictions in complexity impact the translation quality?

On top of the simplification capabilities of the model, also the adequacy of the translations is important. In this section, we investigate the translation quality of the model. We expect that the more we limit the complexity, the less accurate the translation will be. To measure the quality, we use the BLEU (Papineni et al., 2002) metric to compare the output sentences with those in the reference. We present the results in Table 2.

According to the scores, the model trained without tags achieved the highest translation quality. This indicates that it is preferable not to limit the complexity of the output at all. Even the less restricted output (i.e. the N1 output) does not outperform the model with no tags.

Additionally, we validate our hypothesis that constraining the output deteriorates the quality. In the table, the lower the JLPT level is the lower the BLEU scores are. Moreover, we find a significant difference, 9.5 BLEU points, between the highest and lowest level outputs.

As BLEU is an n-gram matching metric, some sentences may convey the same meaning of the reference although they use different spellings (such as the two spellings of "friend" mentioned before). We understand that the reference uses the spelling that is the most comfortable for a Japanese native speaker. However, in the table, we have also included the BLEU scores when both the set of outputs and the reference were converted into hiragana (i.e. column *BLEU (hiragana)*) to avoid mixed spelling.

By doing this, the BLEU scores are higher as there is a higher n-gram overlap with the reference. However, the conclusions are the same: the model without tags performs the best, and the lower the JLPT level the lower the quality is (with 8.6 BLEU points difference between N1 and N5).

6.4 Translation Examples

In the previous section, we introduced that a reason for lower-level outputs to have poor translation quality is due to the lack of information to correctly translate certain terms. For example, as the vocabulary of N5 is more limited, in several sentences of N5 output we find translation mistakes such as wrong translations, or even terms copied directly from the source. We provide examples of these in the following section. Here we present some sentences that illustrate some of the advantages and disadvantages of using tags to control the complexity of the vocabulary. These are included in Table 3.

We see that the word "hat" is translated as "ハット" which is read as "hatto" and corresponds to a transliteration from English using katakana alphabet. For upper levels (i.e. N3 to N1) the terms generated is "帽子" (read as "bōshi") which is written in kanji.

Something similar can be seen with "noses and cheeks". This is translated as "ノーズとチーク" ("Nōzu to chīku") by the N5 output, and it is also closer to a transliteration of the English terms. In the other outputs, this is translated as "鼻と頬", which contain kanji.

Regarding the translation of "companions", in the outputs of upper levels we found "仲間" which is the same as in the reference. The N3 output produces "同行者", which is also spelled in kanji.

Interestingly, the N4 output we find "友だち"

Source	My companions, who weren’t wearing hats, apparently had their noses and cheeks turn red.
Ref	帽子をかぶってなかった仲間は、鼻とほっぺが赤くなっているようでした。
no-tag	帽子を被っていない仲間は、鼻や頬が赤くなっていたそうです。
N1	帽子をかぶっていない仲間は、鼻や頬が赤くなったそうです。
N2	帽子をかぶっていない仲間は、鼻や頬が赤くなったそうです。
N3	帽子をかぶっていなかった私の同行者は、見たところ、彼らの鼻と頬が赤くなりました。
N4	ハットをかぶっていなかった私のお友だちは、鼻と頬が赤くなったようです。
N5	ハットをかぶっていなかったお姉さんは、ノーズとチークが赤くなっていたそうです。
Source	Tom and Mary have gone hunting
Ref	トムとメアリーは狩りに行ったよ。
no-tag	トムとメアリーは狩りに行きました
N1	トムとメアリーは狩りに行った。
N2	トムとメアリーはハンティングに行きました。
N3	トムとメアリーはハンティングに行った。
N4	トムとメアリーはハンティングに行きました。
N5	Tom と Mary メアリー have gone 行った hunting ハンティング

Table 3: Translation Examples.

which means “friend”. As seen in Section 4.1, this word could be written as “友達”. However, only the kanji 友 belongs to N4. The other kanji, “達”, belongs to a higher level than that stated in the input. Therefore, the model produced the hiragana spelling of that part.

In the N5 output, as the tag is the most restrictive, the word “companions” is too complex to be translated. In this case, the word generated is “お姉さん” which means “older sister”, and do not convey the same meaning.

In the second example, we present different ways of how the word “hunting” has been translated by the models. First, “no-tag” and N1 outputs correctly produce the kanji “狩”. This kanji belongs to the N1 level, therefore these are the outputs where we find it. The rest of the outputs produce the term “ハンティング” which is a transliteration, in katakana, of the English term.

Regarding the N5 output, this is another example of how limiting the complexity could harm the translation. Many words have been copied from the source instead of being translated. In this sentence, the word “hunting” has been generated twice: one copied from the English side, and the other one as transliteration.

7 Conclusion and Future Work

In this work, we have used the addition of tags to control the complexity of the output of an English-to-Japanese MT model. The complexity has been established based on the vocabulary and kanji of JLPT exams.

Our results show that the complexity of the lexicon in the translation can be modulated with these tags. Despite that, although it can be influenced to a certain extent, the output may contain vocabulary of higher levels than that stated. This is not only because in the lower levels the vocabulary is too limited, but also because of the lack of translation occurrences in the train data.

We have also shown that restricting the output harms the translation quality. None of the outputs obtained using a complexity tag was better than that of a model trained without any restriction. In addition, enforcing too much simplicity causes the model not to be able to translate accurately and in some cases, it ends up copying words from the source.

One limitation of this work is that the classification of difficulty is decided solely based on the vocabulary. In future work, we want to expand this to also consider other factors such as the grammar or length of the sentences.

Another aspect that we want to investigate is using alternative configurations. For example, text simplification or paraphrasing models (Maddela et al., 2021) could be included to change the distribution of the complexity in the training data.

References

Sweta Agrawal and Marine Carpuat. 2019. Controlling text complexity in neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1549–1564, Hong Kong, China.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–391, Vancouver, Canada.

Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 63–68, Beijing, China.

Muhaimin Hading, Yuji Matsumoto, and Maki Sakamoto. 2016. Japanese lexical simplification for non-native speakers. In *Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016)*, pages 92–96, Osaka, Japan.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia.

Mounica Maddela, Fernando Alva-Manchego, and Wei Xu. 2021. Controllable text simplification with explicit paraphrasing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3536–3553.

Kelly Marchisio, Jialiang Guo, Cheng-I Lai, and Philipp Koehn. 2019. Controlling the reading level of machine translation output. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 193–203, Dublin, Ireland.

Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020. Controllable sentence simplification. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4689–4698, Marseille, France.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2020. JParaCrawl: A large scale web-based English-Japanese parallel corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3603–3609, Marseille, France.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 529–533, Portland, Oregon, USA.

Daiki Nishihara, Tomoyuki Kajiwara, and Yuki Arase. 2019. Controllable text simplification with lexical constraint loss. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 260–266, Florence, Italy.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

Nicholas Ramkissoon. Japanese reading difficulty classification for non-native language learners.

Matthew Shardlow and Fernando Alva-Manchego. 2022. Simple tico-19: A dataset for joint translation and simplification of covid-19 texts. In *Proceedings of the 13th Language Resources and Evaluation Conference*, pages 3093–3102, Marseille, France.

Nicolas Spring, Annette Rios Gonzales, and Sarah Ebling. 2021. Exploring german multi-level text simplification. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1339–1349.

Sanja Štajner and Maja Popović. 2016. Can text simplification help machine translation? In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, pages 230–242, Riga, Latvia.

Sanja Štajner and Maja Popović. 2019. Automated text simplification as a preprocessing step for machine translation into an under-resourced language. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1141–1150, Varna, Bulgaria.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, Long Beach, USA.

Jonathan Waller. 2010. Japanese language proficiency test resources.

Tong Wang, Ping Chen, John Rochford, and Jipeng Qiang. 2016. Text simplification using neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, pages 4270–4270.